# Full-Waveform Airborne LiDAR Data Classification using Convolutional Neural Networks

Stefano Zorzi, Eleonora Maset, Andrea Fusiello and Fabio Crosilla

***Abstract*—Point-cloud classification is one of the most important and time consuming stages of airborne LiDAR data processing, playing a key role in the generation of cartographic products. This paper describes an innovative algorithm to perform LiDAR point-cloud classification, that relies on Convolutional Neural Networks and takes advantage of full-waveform data registered by modern laser scanners. The proposed method consists of two steps. First, a simple CNN is used to pre-process each waveform, providing a compact representation of the data. Exploiting the coordinates of the points associated to the waveforms, output vectors generated by the first CNN are then mapped into an image, that is subsequently segmented by a Fully Convolutional Network: a label is assigned to each pixel and, consequently, to the point falling in the pixel. In this way, spatial positions and geometrical relationships between neighbouring data are taken into account. These particular architectures allow to accurately identify even challenging classes such as power line and transmission tower.**

***Index Terms*—LiDAR · Full-waveform · Classification · Deep learning · Convolutional Neural Network**

## I. Introduction

AIRBORNE laser scanning (ALS) relies on the LiDAR (Light Detection and Ranging) principle, namely to measure the time of flight of a short laser pulse travelling to the target and back, that allows to compute the distance between the sensor and the target. Ranges are then converted to discrete 3D points exploiting GNSS and IMU (Inertial Measurement Unit) data. During its path, the laser ray can be reflected by more than one surface placed at different heights, e.g. part of the laser beam can be reflected from the top of a tree and some part within the tree or the ground surface. The first commercial laser scanners detected only the first and last echo per emitted pulse. Nowadays, most instruments have the ability to record up to six reflections for each emitted pulse and, since 2004, these multi-echo laser scanners have been joined by a new category, the so called *full-waveform* laser scanners, that are finally able to record the entire waveform of the reflected signal. Several studies have shown that these instruments provide a higher spatial point density as well as additional information on the characteristics of the target [1], [2]. In fact, the shape and size of the backscattered waveform is related to the geometry and the reflectance properties of the hit surface.

ALS is currently being employed in a variety of applications, including urban planning, natural hazard management, forestry and facilities monitoring. In almost all the applications, the classification of LiDAR point-cloud is required, being a necessary processing step, e.g., to create Digital Terrain Models (DTMs), to perform analyses on data belonging to particular classes (e.g., to evaluate the vegetation density) and to automatically determine the relationships between different classes (e.g., to calculate the distance between power line conductors and vegetation or buildings).

The aim of this paper is to propose a new classification method for full-waveform airborne LiDAR data using Convolutional Neural Networks (CNNs) and exploiting both full-waveform and spatial information. Thanks to the combination of a first CNN that provides a compact representation of the waveforms, and a subsequent Fully Convolutional Network (FCN) that takes into account also the spatial relations between the points, the proposed network is able to distinguish among (e.g.) six classes, namely: *ground*, *vegetation*, *building*, *power line*, *transmission tower* and *street path*, with an overall accuracy of 92.6%.

The paper is organized as follows. In the next section, the literature on full-waveform LiDAR data is reviewed. Section III describes in detail the proposed method, while Sec. IV introduces the dataset used for the validation and shows the results. Finally, Sec. V draws the conclusion.

## II. Related Work

As shown in several studies [3], [4], [5], [6], the LiDAR point-cloud classification process can significantly benefit from the data collected by full-waveform laser scanners. In fact, the waveform registered by these instruments offers the possibility to extract additional features related to the reflectivity characteristics of the target. Over the last years, several classification methods have been proposed in the literature using full-waveform data and the features derived from them [7]. Among these, we mention decision trees, manually tuned [8] or learned from data [9]. The first method [8] distinguishes between vegetation and non-vegetation points with an overall accuracy of 89.9% for a dense natural forest and 93.7% for a garden area, exploiting the number of echoes, echo width and total cross-section extracted from the waveforms. In [9] the backscatter coefficient is used, along with spatial attributes, to identify flat roofs, pitched roofs, grass, road, trees and shrubs with an overall accuracy of 91.5%. Please note that flat and pitched roofs are subclasses of the common building class, just like trees and shrubs are subclasses of vegetation.

Other methods are based on statistical learning, like Support Vector Machines (SVM) classifiers [10], which belong to

Stefano Zorzi is with the Institute of Computer Graphics and Vision (ICG), Graz University, Inffeldgasse 16, 8010 Graz, Austria. E-mail: stefano.zorzi@icg.tugraz.at

Eleonora Maset, Andrea Fusiello and Fabio Crosilla are with the DPIA, University of Udine, Via Delle Scienze, 206, 33100 Udine, Italy. E-mail: maset.eleonora@spes.uniud.it; (fabio.crosilla; andrea.fusiello)@uniud.it

non-parametric methods and perform non-linear classification. This algorithm is well suited for high dimensional problems with limited training set and proved to reach high accuracy (around 95%) when distinguishing between three classes, namely ground, vegetation and building. For urban vegetation detection Höfle et al. [11] use instead geometric and radiometric features that are fed to an artificial neural network classifier consisting of a single hidden layer of neurons and trained by back propagation. Finally, Wang and Glennie [12] apply a ”voxelization” method that divides the waveform data into voxels, merging the ones falling in the same voxel into a synthesized waveform. Features are then extracted and fused with the information derived from hyperspectral images, constituting the input of a SVM that is able to discriminate between 9 classes with an overall accuracy of 92.6%.

All these algorithms rely on hand-crafted features, that are subsequently fed to statistical classifiers or simple machine learning algorithms. An alternative approach is the one proposed by Maset et al. [13], that exploits a Kohonen's Self Organizing Maps (SOMs) to perform the unsupervised classification of raw full-waveform data without the need of extracting features from them. The method proved to reach an accuracy of 93.1% over three different classes: grass, trees and road.

In the last years disciplines such as computer vision, speech and audio processing, robotics and bioinformatics have pushed forward and exploited the potential of deep learning [14]. Approaches based on hand-engineered features can nowadays be effectively replaced by methods that learn both features and classifier from the data end-to-end. In particular, Convolutional Neural Networks (CNNs) represent a very successful tool for image classification and segmentation [15], [16].

While many researchers are focused on the development of new architectures for image and video processing, the application of deep learning to LiDAR data – and, notably, to full-waveform data – is still almost unexplored.

In the case of conventional LiDAR data, the works of Hu et al. [17] and Yang et al. [18] can be recalled, in which the potential of CNNs for the classification of LiDAR data is demonstrated. More specifically, in [17] a CNN is used to detect ground points, exploiting a point-to-image framework. For each point in the dataset, context information are computed from the neighbouring points in a window and subsequently transformed into an image that is fed to a CNN. In this way, point classification is treated as the binary classification of an image. Similarly, Yang et al. [18] perform a multi-class segmentation of the point-cloud by first transforming the 3D neighbourhood features of a point into a 2D image that is then classified by a CNN. The method reaches an overall accuracy of 82.3% when distinguishing between nine classes, showing however poor performances in the identification of points belonging to small and thin objects such as power line and fences. Recently, Rizaldy et al. [19] proposed an approach based on deep learning for ground classification.

Our system is novel both in the type of data it consumes – full-waveform – and in the approach to the problem. Unlike the aforementioned methods, we treat the LiDAR data classification task as a problem of image segmentation solved with a FCN that takes advantages also on the full-waveform data processed by a CNN classifier.

## III. Proposed Framework

As previously mentioned, the novel method proposed in this paper tries to take advantage of the useful information provided by waveforms recorded by modern laser scanners and of the potentialities offered by deep learning for solving classification and segmentation tasks. The entire architecture is summarized in Figs. 1 and 2 and described in detail in the following sections.

Hyperparameters have been tuned through the typical trial and error approach in order to have a good trade off between accuracy, training time and GPU memory footprint.

### A. Feature Extraction

In the first step of the algorithm, raw waveform data are given as input to a classifier that outputs a vector of length $n$ (with $n$ total number of classes) containing the probability that the analysed input belongs to a certain class. The idea is to train a CNN classifier that provides a compact way to describe each waveform. CNNs are, in fact, a specialized kind of neural network for processing data that have a known grid-like topology, so they can also be applied to time series data such as audio tracks or, as in this case, the recorded waveforms.

The architecture of the CNN used in the proposed method is shown in the upper part of Fig. 2. More in detail, the waveform, consisting of a vector of 160 elements, is fed into two consecutive 1D convolutional layers with kernel size 3, that have 32 and 64 filters, respectively. Both layers are followed by a rectified linear unit (ReLU) activation function and a max-pooling layer with kernel size 2. The activation function is necessary to introduce non-linearity, whereas the max-pooling layer helps to achieve approximate invariance to small translations of the input (due to sloppy windowing of the signal) and to reduce the representation size in the inner layers, which is a hallmark of CNN, thereby decreasing the computational effort.

After the convolutional layers, the network exploits two fully connected layers to do the classification. The number of neurons is 2048 and 1024, respectively. Both fully connected layers are followed by a ReLU activation function and a dropout layer (useful to reduce over-fitting [20]) with a dropout rate of 0.5. The output layer is a $n$ neurons layer followed by *softmax* activation function which produces a probability distribution over $n$ classes.

As an alternative to this model, we tested also various autoencoder configurations to generate a description of the waveform. However, as it will be shown in Sec. IV-C, the classifier proved to perform better.

### B. Point-cloud to Image

The accuracy that can be achieved by the first CNN, that exploits only raw waveform data, is not sufficient, thus additional spatial information must be considered for a precise classification.

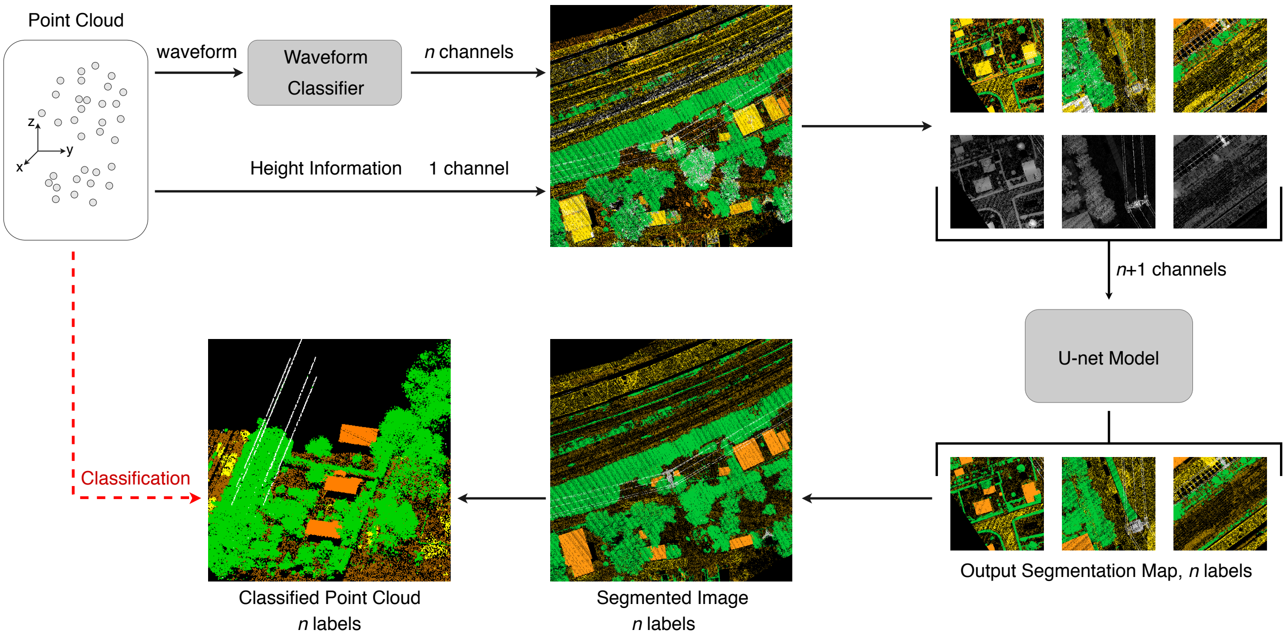


Fig. 1: Workflow of the proposed classification method. First, the waveform classifier (a standard CNN) predicts the point class only exploiting full-waveform data. Predictions are then mapped into an image, together with the height information derived from the 3D coordinates of the points. The resulting multi-channel image is then processed by a FCN (U-net) that refines predictions using spatial information.

The idea is then to map the point-cloud into a two-dimensional orthographic image, exploiting $(x, y)$ coordinates of the points that correspond to the first return (echo) registered in each waveform. In this way, spatial positions and geometrical relationships between neighbouring data are taken into account. The resulting image has multiple channels: every pixel stores the $n$-dimensional probability distribution vector, provided by the classifier employed in the first stage of the procedure, and the height of the data falling in the pixel. The point-cloud classification problem can therefore be cast to the segmentation of an image, that assigns a class label *per-pixel*. This task can be solved by a FCN, as described in detail in Sec. III-C.

Pixel size is adapted to the point-cloud density, however a loss of information inevitably occurs because of *collisions*, i.e., more than one point is mapped to the same pixel. This phenomenon has a negative impact on the accuracy of the algorithm only when involving points of different classes, otherwise a single class label is adequate for all the points. In any case, the point with the highest altitude value is assigned to the pixel, in order to improve classification of small and thin objects such as towers and power lines, which are the most critical classes.

It is possible to limit collisions by reducing the pixel size, which entails enlarging the image, and at the same time increasing the computing time. We used a pixel size of 0.05 m in our experiments, with a rate of collision of approximately 5% but less than 0.5% collisions involve points with different labels.

### C. Image Segmentation via U-net

CNNs were firstly designed to solve image classification tasks, where the desired output is a single class label assigned to the input image. However, in recent years several architectures have been proposed to perform semantic segmentation [21], [22], allowing to assign a class label to each pixel. In particular, we started from the so called U-net model [16] and implemented a FCN to segment the multi-channel image created as described in the previous section. A FCN is composed only of convolutional layers without any fully-connected one. This allows to operate on an input of any size, producing an output of corresponding spatial dimensions [22].

The network we employed, illustrated in Fig. 2, consists of a contracting path (upper part) and an almost symmetrical expansive path (bottom part). In the contracting path, the network looks like a typical CNN able to recognize both low and high level features. Each layer is composed by two $3 \times 3$ convolutions, each followed by batch normalization and ReLU activation function. A $2 \times 2$ max-pooling operation is then applied to reduce the representation size by a factor of two, starting from an input of dimensions $256 \times 256$ and reaching a size of $8 \times 8$ at the final layer of the contracting path. The number of feature channels is doubled at each layer with respect to the previous one. The first layer outputs 64 feature maps, whereas the last one 2048.

Every layer in the expansive path consists instead of an upsampling of the feature maps that increases the resolution of the output of the previous layer, a concatenation with the corresponding feature maps from the contracting path and three $3 \times 3$ convolutions, each followed by batch normalization and a ReLU activation function. At the final layer a $1 \times 1$

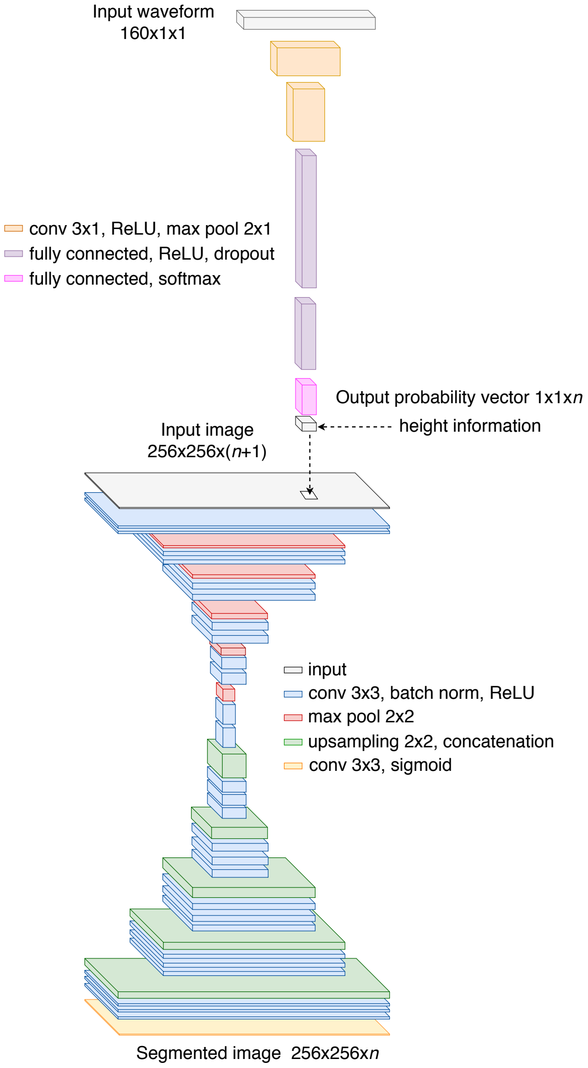

Input waveform
160x1x1
▭ conv 3x1, ReLU, max pool 2x1
▭ fully connected, ReLU, dropout
▭ fully connected, softmax
Output probability vector 1x1xn
height information
Input image
256x256x(n+1)
▭ input
▭ conv 3x3, batch norm, ReLU
▭ max pool 2x2
▭ upsampling 2x2, concatenation
▭ conv 3x3, sigmoid
Segmented image 256x256xn

Fig. 2: Architecture of the proposed networks. At the top, the waveform classifier. At the bottom, the U-net model used for the image segmentation (best viewed in color).

convolution is used to map each 64 components feature vector to the desired number of classes. While the contracting path captures context information, the expansive path enables precise localization [16], thus allowing a *per-pixel* labelling.

The U-net consumes the multi-channel image created as described in Sec. III-B. The first layer of the U-net model is designed so as to take in input images of fixed size (256 × 256 in our case) but a point-cloud can be mapped in a much larger image. An image of arbitrary size can be processed by an overlap-tile strategy. Since convolutions in our U-net are padded, the *valid* portion of the 256 × 256 output layer is reduced by 14 pixels at each side. Therefore input tiles must overlap (by 28 pixels) in order to provide a valid output for each pixel.

## IV. Experiments and Results

The networks have been implemented in Keras [23] and run on a Tesla K40c GPU. Validation has been performed on a dataset that we manually labelled and made available on the web[1] to allow for future comparisons.

### A. Dataset

Our networks have been trained and validated using a dataset acquired by Helica s.r.l. with a Riegl LMS-Q780 full-waveform airborne laser scanner. The surveyed area contains both natural surfaces such as ground and vegetation, as well as artificial objects such as buildings, power lines and transmission towers.

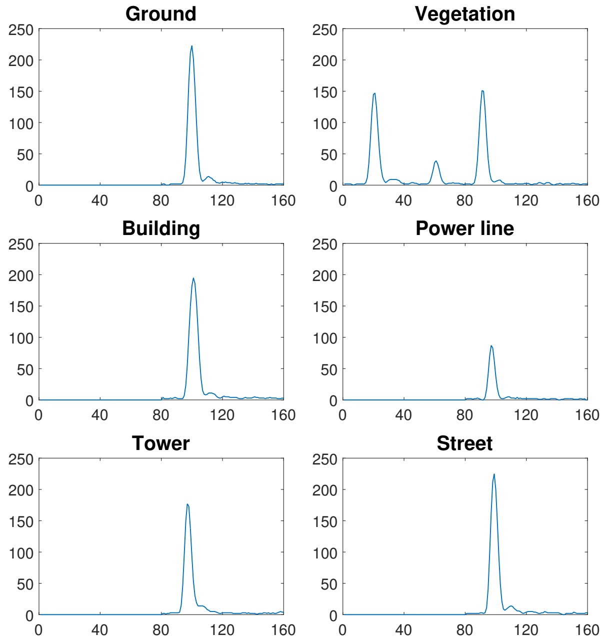


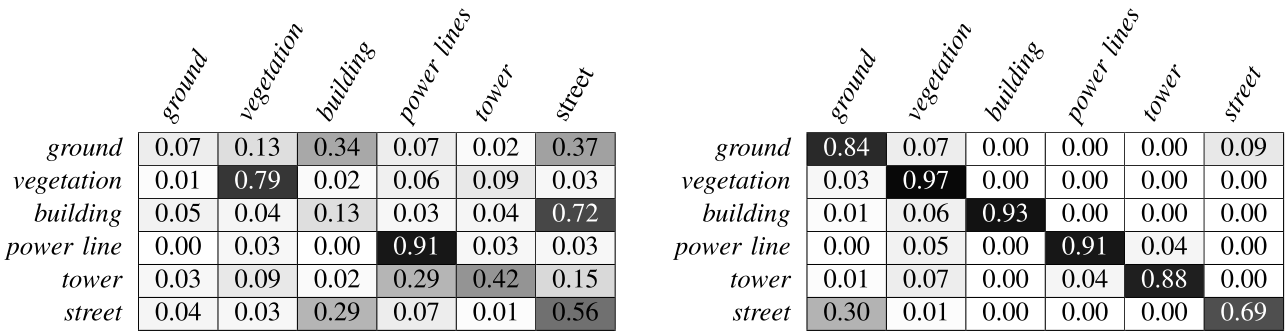
Fig. 3: Waveform samples (zero-padded).

Three different information are associated to every measured point contained in the dataset, namely the waveform registered by the LiDAR full-waveform sensor, described by a vector of 160 values (shortest signals are padded with zeros to reach this length), the 3D coordinates of the point and the label that shows the class to which the point belongs. These labels have been assigned manually among six classes that were identified: *ground*, *vegetation*, *building*, *power line*, *transmission tower* and *street path*.

The point-cloud is composed by more than 9.8 million points, unevenly distributed over the classes. The dataset is indeed very imbalanced due to the different shape of the scanned objects and the occupied area: e.g., the number of points belonging to vegetation and ground is much higher than the number of points belonging to power line and transmission tower classes. Table I shows in detail the points distribution over the classes.

To handle the entire point-cloud, the dataset is divided into subsets, each containing a different number of points. In the experiments, one subset is used as test dataset (corresponding to approximately 10% of the total number of points), while the remaining ones are exploited to train the models.

[1] http://www.dpia.uniud.it/fusiello/demo/fwl/



|  | *ground* | *vegetation* | *building* | *power lines* | *tower* | *street* |
|---|---|---|---|---|---|---|
| *ground* | 0.07 | 0.13 | 0.34 | 0.07 | 0.02 | 0.37 |
| *vegetation* | 0.01 | 0.79 | 0.02 | 0.06 | 0.09 | 0.03 |
| *building* | 0.05 | 0.04 | 0.13 | 0.03 | 0.04 | 0.72 |
| *power line* | 0.00 | 0.03 | 0.00 | 0.91 | 0.03 | 0.03 |
| *tower* | 0.03 | 0.09 | 0.02 | 0.29 | 0.42 | 0.15 |
| *street* | 0.04 | 0.03 | 0.29 | 0.07 | 0.01 | 0.56 |

|  | *ground* | *vegetation* | *building* | *power lines* | *tower* | *street* |
|---|---|---|---|---|---|---|
| *ground* | 0.84 | 0.07 | 0.00 | 0.00 | 0.00 | 0.09 |
| *vegetation* | 0.03 | 0.97 | 0.00 | 0.00 | 0.00 | 0.00 |
| *building* | 0.01 | 0.06 | 0.93 | 0.00 | 0.00 | 0.00 |
| *power line* | 0.00 | 0.05 | 0.00 | 0.91 | 0.04 | 0.00 |
| *tower* | 0.01 | 0.07 | 0.00 | 0.04 | 0.88 | 0.00 |
| *street* | 0.30 | 0.01 | 0.00 | 0.00 | 0.00 | 0.69 |

Fig. 4: Confusion matrices: each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class. Values are normalized so that the sum of every row is equal to 1. Left: output of the waveform classifier (first stage). Right: Output of the U-net (second stage).

TABLE I: Points distribution over the six classes, divided into training and test sets.

|  |  | TRAINING |  | TEST |  |
|---|---|---|---|---|---|
| **Label** | **Class** | **# Points** | **%** | **# Points** | **%** |
| 1 | *ground* | 1787352 | 20.4 | 193070 | 18.1 |
| 2 | *vegetation* | 4719634 | 53.9 | 765327 | 71.7 |
| 3 | *building* | 1514486 | 17.3 | 49138 | 4.6 |
| 4 | *power line* | 71978 | 0.8 | 8151 | 0.8 |
| 5 | *tower* | 32008 | 0.4 | 1829 | 0.2 |
| 6 | *street path* | 633606 | 7.2 | 49580 | 4.6 |

### B. Training

To overcome the imbalanced distribution of the points over the six classes, when training the waveform classifier (Sec. III-A) we sample with replacement a fixed number of waveforms for each class. More specifically, we employ 200 thousand waveforms per class, for a total of 1.2 million samples. We tested also techniques to balance the class distribution for the training stage [24], [25] but no significant improvement on the final results can be noticed.

The training is performed using categorical cross-entropy as loss function and Adam optimizer [26] with 0.001 learning rate, while dropout is applied with rate 0.5 on the two fully-connected layers. The weights are initialized as described in [27]. The CNN has 12 million trainable parameters and, fixing the batch size to 256, a training epoch takes approximately 30 seconds and it converges after a few minutes.

Regarding the U-net (Sec. III-B), the training is done using 15 thousand $256 \times 256$ windows with 7 channels for each pixel (see Fig. 5). Six channels correspond to the probability vector over the six classes provided as output by the classifier, and one channel contains the height information. Please note that the training images are randomly cut out and extracted from the much larger image in which the training point-cloud is mapped. To take into account the unbalancing of the point distribution over the classes, it is ensured that 1700, 3400 and 3400 training images contain pixels belonging to *building*, *power line* and *transmission tower*, respectively, which are the under-represented classes.

For the training of this FCN, categorical cross-entropy is used as loss function and Adam optimizer [26] is applied with learning rate 0.0002, while the weights are initialized as described in [27]. Choosing a batch size of 8 images, the training of the U-net model (with 138 million trainable parameters) takes approximately 80 minutes per training epoch, reaching convergence after 30 epochs.

### C. Testing

In order to report results that are independent from the training stage, to some extent, five trainings were performed independently, each time reinitializing the weights from scratch and randomly extracting the training dataset from the entire point-cloud, as described in Sec. IV-B. The resulting overall accuracy, computed on the test set, is equal to $92.6(\pm 0.7)\%$, while the average per class accuracy is $87.0(\pm 0.3)\%$.

As can be noticed from the confusion matrix represented in Fig. 4 (right), that reports the results for one out of the five trainings, the network performs very well for the classes *vegetation*, *building*, *power line* and *transmission tower*. Instead, points belonging to the class *street path* are often confused with the class *ground*. This is probably due to the fact that the shape of the waveforms belonging to these two classes are often indistinguishable (see Fig. 3) and also the geometric characteristics of *ground* and *street path* points can be very similar. In practical applications (e.g. for the creation of DTMs) these two classes are usually merged together. If we consider *ground* and *street path* as a unique class, the overall accuracy increases to $96.1(\pm 0.2)\%$ and the average per class accuracy to $92.5(\pm 0.5)\%$.

TABLE II: Synopsis of state-of-the-art methods.

| **Ref** | **# classes** | **Method** | **Accuracy** |
|---|---|---|---|
| Ours | 6 | CNN | 92.6 |
| Ours | 5 | CNN | 96.1 |
| [8] | 2 | Dec. Tree | 89.9 - 93.7 |
| [9] | 4(6) | Dec. Tree | 91.5 |
| [10] | 3 | SVM | 95.3 |
| [12] | 9 | SVM | 92.6 (+ hyperspectral) |
| [13] | 3 | SOM | 93.1 |

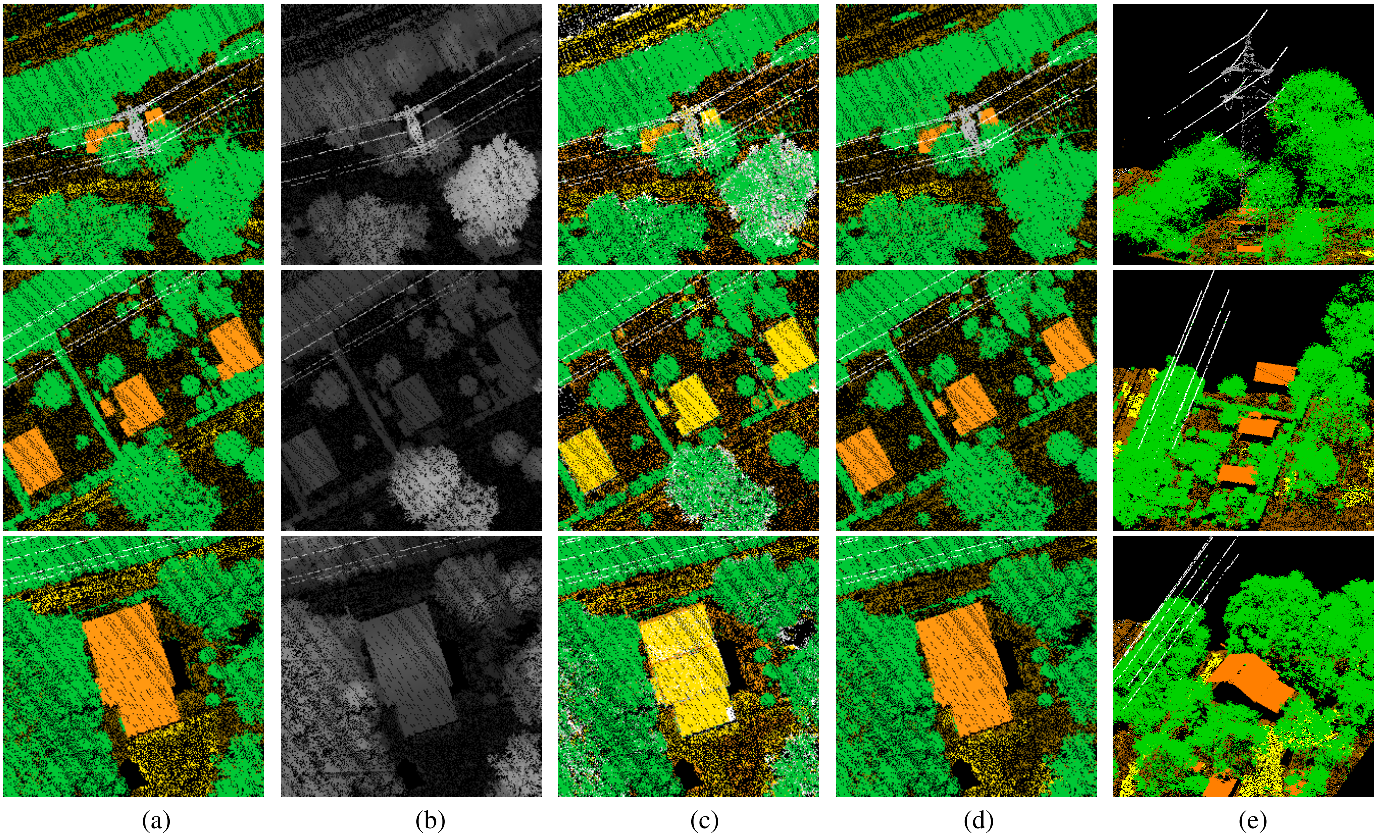


Fig. 5: Sample images ($256 \times 256$) and results (best viewed in colour). (a) Ground truth images used for training and validation; (b) Height channel; (c) Labels predicted by the waveform classifier (maximum probability) that are fed to the U-net; (d) Labels produced by the U-net (maximum probability); (e) 3D views of the classified point-cloud, coloured with the predicted labels. Classes: *ground* (brown), *vegetation* (green), *building* (orange), *power line* (white), *transmission tower* (grey), *street path* (yellow).

Although a direct comparison with other methods using full-waveform LiDAR is not possible, for the labelled full-waveform data used in our experiments is the first public dataset of this kind, Tab. II suggests that our method compares favourably with the state of the art (the table refers to the methods described in Sec. II).

Examples of the input provided to the U-net model and of the obtained results for the test set are shown in Fig. 5.

*a) Ablation study:* We also tested the performances of the waveform classifier alone (Sec. III-A), which turns out to be unsatisfactory, for it reaches 61.1% overall accuracy in the test set. The confusion matrix shown in Fig. 4 (left) indicates that some classes are merged together, namely *ground*, *building* and *street path*, and also the class *transmission tower* is often misclassified. This confirms that our approach reaches high accuracy in the point-cloud classification thanks to the combination of full-waveform data and spatial support.

As previously mentioned, we tried to replace the waveform classifier with autoencoders with different code dimensions. The best performance was achieved with code dimensions equal to the number of classes, but the overall accuracy was only $84.9\%$. When merging the classes *ground* and *street path*, the overall accuracy increases to $89.5\%$.

## V. Conclusion

In this paper we presented an innovative algorithm based on CNNs to perform full-waveform LiDAR point-cloud classification. The proposed network employs directly the raw full-waveform data, learning both features and classifier end-to-end, unlike other methods that require preliminary extraction of features. It can be applied to the classification of points belonging to any kind of area and no prior knowledge on the data characteristics is required.

Experiments reports an overall accuracy of 92.6% on six classes, including challenging instances such as *power line* and *transmission tower*. Although a direct comparison with other methods using full-waveform LiDAR is not possible, experiments suggest that our method compares favourably with the state of the art. The labelled dataset that we made available to the public domain allows reproducibility and comparison by other authors.

*Acknowledgements:* The authors are grateful to Helica s.r.l. for providing the full-waveform LiDAR data and allowing to distribute it in the public domain. They are indebted with A. Abramo for kindly granting access to the Tesla GPU. A patent had been applied for on the research reported in this paper.

## References

[1] W. Wagner, A. Ullrich, T. Melzer, C. Briese, and K. Kraus, "From single-pulse to full-waveform airborne laser scanners: potential and practical challenges," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 35, pp. 201–206, 2004.

[2] C. Mallet and F. Bretar, "Full-waveform topographic lidar: State-of-the-art," *ISPRS Journal of photogrammetry and remote sensing*, vol. 64, no. 1, pp. 1–16, 2009.

[3] V. Ducic, M. Hollaus, A. Ullrich, W. Wagner, and T. Melzer, "3d vegetation mapping and classification using full-waveform laser scanning," in *Proc. Workshop on 3D Remote Sensing in Forestry. EARSeL/ISPRS, Vienna, Austria, 1415 February 2006*, 2006, pp. 211–217.

[4] C. Mallet, F. Bretar, and U. Soergel, "Analysis of full-waveform lidar data for classification of urban areas," *Photogrammetrie Fernerkundung Geoinformation*, vol. 5, pp. 337–349, 2008.

[5] A. L. Neuenschwander, L. A. Magruder, and M. Tyler, "Landcover classification of small-footprint, full-waveform lidar data," *Journal of applied remote sensing*, vol. 3, no. 1, p. 033544, 2009.

[6] J. Reitberger, P. Krzystek, and U. Stilla, "Benefit of airborne full waveform lidar for 3d segmentation and classification of single trees," in *ASPRS 2009 Annual Conference*, 2009, pp. 1–9.

[7] K. D. Fieber, I. J. Davenport, J. M. Ferryman, R. J. Gurney, J. P. Walker, and J. M. Hacker, "Analysis of full-waveform lidar data for classification of an orange orchard scene," *ISPRS journal of photogrammetry and remote sensing*, vol. 82, pp. 63–82, 2013.

[8] W. Wagner, M. Hollaus, C. Briese, and V. Ducic, "3d vegetation mapping using small-footprint full-waveform airborne laser scanners," *International Journal of Remote Sensing*, vol. 29, no. 5, pp. 1433–1452, 2008.

[9] C. Alexander, K. Tansey, J. Kaduk, D. Holland, and N. J. Tate, "Backscatter coefficient as an attribute for the classification of full-waveform airborne laser scanning data in urban areas," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, no. 5, pp. 423–432, 2010.

[10] C. Mallet, F. Bretar, M. Roux, U. Soergel, and C. Heipke, "Relevance assessment of full-waveform lidar data for urban area classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 6, pp. S71–S84, 2011.

[11] B. Höfle, M. Hollaus, and J. Hagenauer, "Urban vegetation detection using radiometrically calibrated small-footprint full-waveform airborne lidar data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 67, pp. 134–147, 2012.

[12] H. Wang and C. Glennie, "Fusion of waveform lidar data and hyperspectral imagery for land cover classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 108, pp. 1–11, 2015.

[13] E. Maset, R. Carniel, and F. Crosilla, "Unsupervised classification of raw full-waveform airborne lidar data by self organizing maps," in *International Conference on Image Analysis and Processing*. Springer, 2015, pp. 62–72.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[17] X. Hu and Y. Yuan, "Deep-learning-based classification for dtm extraction from als point cloud," *Remote sensing*, vol. 8, no. 9, p. 730, 2016.

[18] Z. Yang, W. Jiang, B. Xu, Q. Zhu, S. Jiang, and W. Huang, "A convolutional neural network-based 3d semantic labeling method for als point clouds," *Remote Sensing*, vol. 9, no. 9, p. 936, 2017.

[19] A. Rizaldy, C. Persello, C. M. Gevaert, and S. J. O. Elberink, "Fully convolutional networks for ground classification from lidar point clouds," *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 4, no. 2, pp. 231–238, 2018.

[20] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[21] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in neural information processing systems*, 2012, pp. 2843–2851.

[22] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[23] F. Chollet, "Keras," https://github.com/ fchollet/keras, 2015.

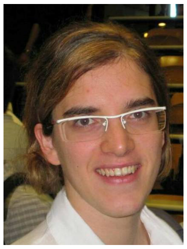
[24] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[25] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge & Data Engineering*, no. 9, pp. 1263–1284, 2008.

[26] D. Kinga and J. B. Adam, "A method for stochastic optimization," in *International Conference on Learning Representations*, vol. 5, 2015.

[27] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

**Stefano Zorzi** received the B.Sc. and M.Sc. degrees in Electrical Engineering from the University of Udine, Udine, Italy, in 2015 and 2017, respectively. Since June 2018, he is a Ph.D. student at the Institute of Computer Graphics and Vision (ICG), Graz University, Austria. His current research interests include computer vision and image analysis. In particular, he is dealing with semantic segmentation, classification and object detection in images using deep learning techniques.



**Eleonora Maset** received the M.Sc. degree in Environmental Engineering and the Ph.D. in Industrial and Information Engineering from the University of Udine, Udine, Italy, in 2015 and 2019, respectively. She is currently Postdoc at the Polytechnic Department of Engineering and Architecture (DPIA) at University of Udine and collaborates with Helica srl. Her research interests include laser scanning, computer vision and image analysis.

**Andrea Fusiello** received the Laurea (M.S.) degree in computer science from the University of Udine, Udine, Italy, and the Dottorato di Ricerca (Ph.D.) degree in computer engineering from the University of Trieste, in 1994 and 1999, respectively. He was a Research Fellow with Heriot-Watt University, Edinburgh, in 1999. From 2001 to 2011, he was with the Department of Computer Science, University of Verona. As an Associate Professor in 2012 he joined the DPIA at the University of Udine, where he is involved in teaching computer vision and computer science basics. In 2016 he as been appointed as co-chair of WG II/1 of the ISPRS. His current research interests include computer vision, image analysis, 3-D model acquisition, and image-based rendering.

**Fabio Crosilla** is Full Professor of Photogrammetry at the Polytechnic Department of Engineering and Architecture (DPIA) at University of Udine, Udine, Italy, and lecturer of Surveying, Digital Mapping and GIS at the same University. Alexander von Humboldt Foundation fellow, he spent research periods at the Geodetic Institute of Stuttgart University in 1984 and 1985. He served for many years in the International Society for Photogrammetry and Remote Sensing. In the period 2010–2014 he acted as Italian Prime Delegate in the European Spatial Data Research network (EuroSDR). He is author of 2 patents and more than 220 publications, of which more than one hundred on applied statistics in Surveying and Photogrammetry.